First published in *Journal of Applied Probability* **45**(2), 531-541.

LAPLACE TRANSFORMS OF PROBABILITY DISTRIBUTIONS AND THEIR INVERSIONS ARE EASY ON LOGARITHMIC SCALES

A. G. ROSSBERG,* *IIASA*

Abstract

It is shown that, when expressing arguments in terms of their logarithms, the Laplace transform of a function is related to the antiderivative of this function by a simple convolution. This allows efficient numerical computations of moment-generating functions of positive random variables and their inversion. The application of the method is straightforward, apart from the necessity to implement it in high-precision arithmetics. In numerical examples the approach is demonstrated to be particularly useful for distributions with heavy tails, such as lognormal, Weibull, or Pareto distributions, which are otherwise difficult to handle. The computational efficiency compared to other methods is demonstrated for a M/G/1 queuing problem.

Keywords: Moment generating function, Laplace transform, transform inversion, heavy tails

2000 Mathematics Subject Classification: Primary 44A10

Secondary 33F05; 44A35; 33B15

1. Introduction

Moment-generating functions (m.g.f.) are frequently used in probability theory. However, computing a m.g.f. from a given distribution, and, even more so, computing a distribution from a given m.g.f., can be challenging. Here, a new numerical method for these transformations is proposed. The method is particularly powerful for distributions that are well-behaved on a logarithmic scale, e.g., for lognormal and other heavy-tailed distributions. Sums of such distributions are encountered in analyses of problems as diverse as radio communication [4, 23, 7, 8], tunnel junctions [20], turbulence [12], biophysics [17, 28], and finance [18, 11]. Sums of lognormals have so far been difficult

* Postal address: Evolution and Ecology Program, International Institute for Applied Systems Analysis (IIASA), Schlossplatz 1, 2361 Laxenburg, Austria

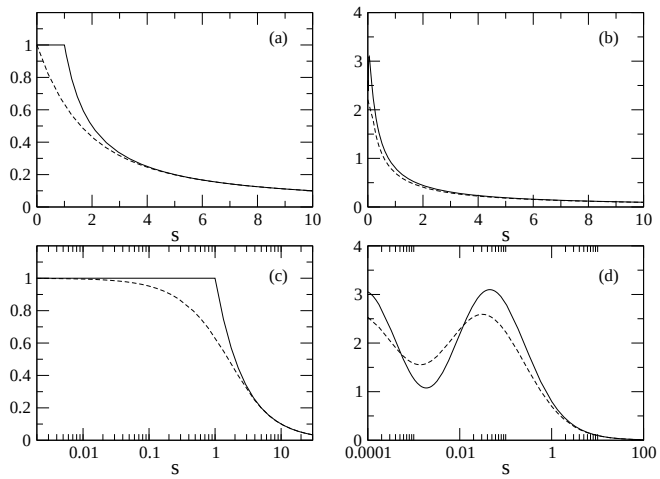

FIGURE 1: Examples comparing the integral $\int_0^{1/s} f(x)\,dx$ (solid) with the Laplace transform (dashed) of functions $f(x)$. (a): $f(x) = 1$ for $x < 1$, $f(x) = 0$ otherwise, (b): $f(x) = \sin[\ln(x+1)]/x$, (c) and (d): same as (a) and (b), but on semi-logarithmic axis.

to handle [4, 6, 18, 20]. Since the m.g.f. is a variant of the Laplace transform, and since the theory applies to Laplace transforms in general, I shall first introduce it in this framework, and discuss its application in probability theory, including numerical examples, later on.

As shown for two examples in Figs. 1a,b, the graphs of the Laplace transform

$$[\mathcal{L}f]\,(s) = \int_0^\infty e^{-st} f(t)\,dt = F(s) \tag{1}$$

of a function $f(t)$ and of its integral $\int_0^{1/s} f(t)\,dt$ with reciprocal upper bound $s > 0$ look quite similar. The similarity becomes even more striking when going over to logarithmic scales, i.e., when comparing $H(y) := F(e^y)$ with $h(y) := \int_0^{e^{-y}} f(t)\,dt$ (see Figs. 1c,d). In fact, as is shown below, $H(y)$ can be obtained from $h(y)$ by a simple convolution. Conversely, $h(y)$, and therefore the inverse of the Laplace transform, can be computed from $H(y)$ by a deconvolution, which can efficiently be implemented numerically using Fast Fourier Transforms (FFT).

The problem of numerically inverting Laplace transforms has persistently attracted attention. Valkó and Vojta [27] compiled a list of over 1500 publications devoted to it dating from 1795 to 2003. Important methods used today are, for example, the Gaver-Stehfest method [13, 24] and its variants [26], the Euler Algorithm [9], and the

Talbot Algorithm [25]. Abate and Whitt recently compared these methods within a generalized formal framework [3].

2. General Theory

2.1. Main theorem

To see that, on a logarithmic scale, Laplace transform and integral of a function are related by a convolution, first recall a simple fact not always stated precisely in the textbooks:

Lemma 1. *Let $f : \mathrm{R}^{\geq 0} \to \mathrm{R}$ (or a corresponding linear functional) and assume its Laplace transform $[\mathcal{L}f](s)$ to be defined for all $s > 0$. Define $g(x) := \int_0^x f(t)\,dt$. Then*

$$[\mathcal{L}f](s) = s\,[\mathcal{L}g](s) \tag{2}$$

for all $s > 0$.

Proof. Formally, this is, of course, just an integration by parts:

$$\int_0^T e^{-st} f(t)\,dt = e^{-sT} \int_0^T f(t)\,dt + s \int_0^T e^{-st} g(t)\,dt. \tag{3}$$

Subtle is only the question if the boundary term vanishes as $T \to \infty$. It does, because

$$\begin{aligned} \left| e^{-sT} \int_0^T f(t)\,dt \right| &= e^{-sT/2} \left| \int_0^T e^{-sT/2} f(t)\,dt \right| \\ &< e^{-sT/2} \left| \int_0^T e^{-st/2} f(t)\,dt \right|, \end{aligned} \tag{4}$$

and the last integral converges to $[\mathcal{L}f](s/2)$ as $T \to \infty$, which is finite by assumption, while the exponential factor goes to zero. Thus, Eq. (3) converges to Eq. (2) as $T \to \infty$.

It follows immediately that

Theorem 1. *Under the conditions of Lemma 1, with $h(y) := g(e^{-y})$, $H(y) := [\mathcal{L}f](e^y)$ and $K(y) := \exp(-e^y)\,e^y$ for all $y \in \mathbf{R}$,*

$$H = K * h, \tag{5}$$

where $*$ denotes the convolution operator.

Note that, in a less precise but more transparent notation, Eq. (5) reads

$$[\mathcal{L}f](e^y) = [\exp(-e^y)\, e^y] * \int_0^{e^{-y}} f(z)\, dz. \tag{6}$$

Proof. Apply Lemma 1 and then change variables as $T = e^{-x}$:

$$\begin{aligned} H(y) = e^y\, [\mathcal{L}g](e^y) &= e^y \int_0^\infty \exp(-e^y\, T)\, g(T)\, dT \\ &= \int_{-\infty}^\infty \exp(-e^{y-x})\, e^{y-x}\, g(e^{-x})\, dx \\ &= \int_{-\infty}^\infty K(y-x) h(x)\, dx = [K * h](y). \end{aligned} \tag{7}$$

2.2. Representation in Fourier space

The convolution in Eq. (5) and its inversion are efficiently computed in Fourier space. Define for any function $g(x)$ its Fourier transform $\tilde{g}(k)$ such that

$$\tilde{g}(k) = \int_{-\infty}^\infty e^{-ikx} g(x)\, dx,\; g(x) = \int_{-\infty}^\infty \frac{e^{ikx}}{2\pi} \tilde{g}(k)\, dk. \tag{8}$$

($\tilde{g}(k)$ might have to be interpreted as a linear functional.) Then

$$\tilde{H}(k) = \tilde{K}(k)\tilde{h}(k). \tag{9}$$

The Fourier transform of the kernel can be obtained by a change of variables $u = e^y$ as

$$\begin{aligned} \tilde{K}(k) &= \int_{-\infty}^\infty K(y) e^{-iky}\, dy \\ &= \int_{-\infty}^\infty \exp\left(-e^y\right) e^y e^{-iky}\, dy \\ &= \int_0^\infty e^{-u} u^{-ik} du \\ &= \Gamma(1-ik). \end{aligned} \tag{10}$$

Thus

$$\tilde{h}(k) = \frac{\tilde{H}(k)}{\Gamma(1-ik)}. \tag{11}$$

At this point it is interesting to note that $1/\Gamma(1+z)$ is an entire function [19] and that the coefficients d_n of its Taylor series

$$\frac{1}{\Gamma(1+z)} = \sum_{n=0}^{\infty} d_n z^n \tag{12}$$

are known to be given by $d_0 = 1$ and the recursion [14]

$$(n+1)d_{n+1} = \gamma_E d_n + \sum_{k=1}^{n} (-1)^k \zeta(k+1) d_{n-k} \tag{13}$$

for $n \geq 0$, where γ_E is Euler's constant and $\zeta(x) = \sum_{n=1}^{\infty} n^{-x}$ denotes Riemann's zeta function. The coefficients d_n have been shown to decay to zero faster than $(n!)^{-(1-\epsilon)}$ for any $\epsilon > 0$ [16]. This might sometimes be sufficiently fast to obtain a convergent series representation of $h(x)$ as

$$\begin{aligned} h(x) &= \frac{1}{2\pi} \int_{-\infty}^{\infty} e^{ikx} \sum_{n=0}^{\infty} d_n (-ik)^n \tilde{H}(k)\, dx \\ &= \sum_{n=0}^{\infty} d_n \left(-\frac{d}{dx}\right)^n H(x). \end{aligned} \tag{14}$$

Numerically, this expansion has already been demonstrated to yield accurate results [21]. It would be interesting to understand under which conditions and at what computational cost this point-wise Laplace inversion formula will generally converge.

2.3. Application to probability theory

The moment-generating function $M(t)$ of a real-valued random variable U with cumulative distribution (c.d.f.) $P(u) = P[U \leq u]$ is defined as the expectation value $M(t) = \mathsf{E}\, e^{tU} = \int_{-\infty}^{\infty} e^{tu}\, dP(u)$. Let $p(u) = dP(u)/du$ denote the probability density of U (possibly defined in the functional sense) and assume U to attain only positive values, i.e., $p(u) = 0$ for $u \leq 0$. Then, obviously, $M(-t) = [\mathcal{L}\, p](t)$. Note that $M(-t) = \mathsf{E}\, e^{-tU} \leq 1$ for $t \geq 0$, even if all moments of U are undefined. Hence, the corresponding Laplace transform is always defined for positive arguments, and above considerations apply with

$$f(u) = p(u), \quad h(y) = P(e^{-y}), \quad \text{and} \quad H(y) = M(-e^{y}). \tag{15}$$

Equation (5) then becomes

$$M(-e^{y}) = K(y) * P(e^{-y}). \tag{16}$$

Invertibility of the Laplace transform of $p(u)$ implies that knowledge of $M(t)$ for negative arguments is sufficient to recover $p(u)$. The standard procedure to compute moments of U directly from derivatives of $M(t)$ at $t = 0$ turns out to be numerically difficult when using Eq. (5), but efficient methods exist to compute moments of the logarithm of U directly from $M(t)$ [21]. These can be used, among others, to construct lognormal approximations of a random variable from a given m.g.f..

The fact that a convolution of densities on logarithmic scales corresponds to a multiplication of r.v. implies a close relationship between multiplication and Laplace transforms, which is reflected by the following:

Theorem 2. *Let X and Λ be two independent r.v. such that X is always positive and Λ is exponentially distributed with $\mathsf{E}\Lambda = 1$. Denote by M_X the m.g.f. of X and by P_Z the c.d.f. of*

$$Z := \Lambda / X. \tag{17}$$

Then

$$M_X(-t) = 1 - P_Z(t) \tag{18}$$

for any $t > 0$.

Proof. Define $z = \ln Z$, $\lambda = \ln \Lambda$ and $x = \ln X$, and denote by $P_X(t)$ the c.d.f. of X. Taking logarithms on both sides of Eq. (17) yields $z = \lambda + (-x)$. The density of λ equals $K(\lambda)$, since $\exp(-\Lambda)d\Lambda = \exp(-e^{\lambda})e^{\lambda}d\lambda = K(\lambda)d\lambda$. The c.d.f. of $-x$ is $1 - P_X(e^{-x})$. By the rule for the distribution of sums of independent r.v., the c.d.f. of z, i.e., $P[z < y] = P_Z(e^y)$, equals $K(y) * [1 - P_X(e^{-y})] = 1 - K(y) * P_X(e^{-y})$. Thus, by Eq. (16), $1 - P_Z(e^y) = K(y) * P_X(e^{-y}) = M_X(-e^y)$, which proves Eq. (18) with $y = \ln t$.

3. Numerical Examples

3.1. General considerations

For the application to probability theory described above, it is not difficult to see that $h(y), H(y) \to 0$ as $y \to \infty$ and $h(y), H(y) \to 1$ as $y \to -\infty$. Thus, for any

numerical range of integration $[y_0, y_1]$, direct FFTs of $h(y)$ or $H(y)$ would lead to artifacts because these functions are not periodic.

With sufficiently small y_0 and sufficiently large y_1, approximate periodicity can, for example, be achieved by splitting $h(y) = h_1(y) + h_0(y)$, with

$$h_0(y) = \frac{y - y_1}{y_0 - y_1}. \tag{19}$$

$H = K * h$ is then obtained as the sum $H(y) = H_1(y) + H_0(y)$, where $H_1 = K * h_1$ is computed numerically using FFTs over $[y_0, y_1]$ and

$$H_0(y) = [K * h_0](y) = \frac{y - y_1 + \gamma_E}{y_0 - y_1}. \tag{20}$$

The calculations used to evaluate the numerical examples hereafter are therefore characterized by four parameters: the lower y_0 and upper y_1 end of the interval taken into account, the number N of equally spaced mesh points in this interval at which $h(y)$ and $H(y)$ are computed, and the numerical accuracy ϵ at which computations are done. High-precision arithmetics is needed, just as for other methods of Laplace-Transform inversion [1].

While the implementation of the algorithm is straightforward when an arbitrary-precision FFT library is available, there is no systematic method, yet, for setting the parameters to achieve a desired accuracy of the output. Things to keep in mind when choosing the parameters are that $-y_0$ and y_1 need to be large enough to avoid aliasing and N must be sufficiently large to resolve $h(y)$ on the scale $(y_1 - y_0)/N$. For an appropriate choice of ϵ, note the following: After performing the desired manipulations of $H(y)$, one obtains another generating function H' for which the corresponding distribution is to be computed. The deconvolution of $H'(y)$ (or $H_1'(y)$) to obtain $h'(y)$ (or $h_1'(y)$) is, as any deconvolution, sensitive to numerical errors in $H'(y)$. A simple rule for suppressing artifacts from such errors is to set all Fourier modes of $\tilde{h}_1'(k) = \tilde{H}_1'(k)/\tilde{K}(k)$ to zero that lie beyond the absolute minimum of the power spectrum of $h_1'(y)$, estimated by some local averaging of $|\tilde{h}_1'(k)|^2$. This procedure is justified, because smoothness arguments suggest that $|\tilde{h}_1'(k)|^2 \to 0$ as $|k| \to \infty$. The numerical value of the minimum power gives a coarse estimate of the achievable accuracy (squared), and can be used to tune ϵ.

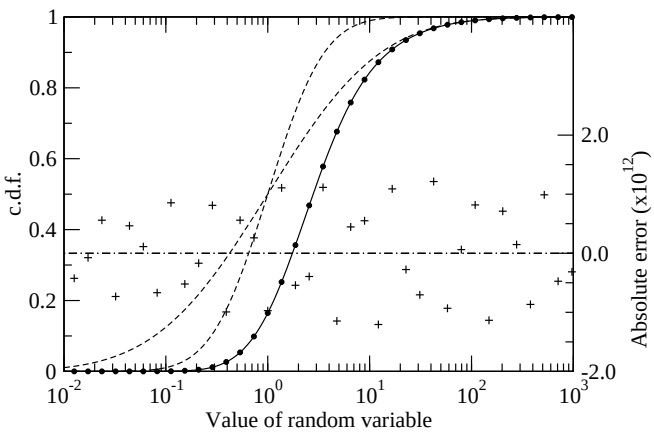

FIGURE 2: The c.d.f. of the sum $e^{\xi_1}+e^{2\xi_2}$ with independent standard normal ξ_1, ξ_2, computed using the methods describe here. Dots correspond to the numerical grid, the solid line is a 5th order spline interpolation. Dashed lines represent the c.d.f. of the two addends. Absolute numerical errors are indicated by crosses.

3.2. Sum of two lognormals

As a first simple example, the c.d.f. of the sum of two lognormal random variables e^{ξ_1}, $e^{2\xi_2}$ with independent standard normal ξ_1 and ξ_2 is computed. The m.g.f. of the sum is obtained as the product of the m.g.f. of the two distributions. Using the method described above with $-y_0 = y_1 = 40$, $N = 256$, and $\epsilon = 10^{-20}$, the c.d.f. of the sum can be computed to a numerical accuracy better than $2 \cdot 10^{-12}$, as is illustrated in Fig. 2. The precise c.d.f. used for comparison was obtained by a direct numerical convolution of the density of e^{ξ_1} with the c.d.f. of $e^{2\xi_2}$ using a high-precision integration algorithm. The comparison shows that the numerical error is of the same order of magnitude at any point of the c.d.f.. This observation suggests that the accuracy at which the c.d.f. and the m.g.f. were computed by the algorithm can be estimated *a posteriori* by the precision at which $H(y)$ respectively $h(y)$ converge to 0 or 1 in the tails.

3.3. Sums of 111 Weibull Distributions

The next two examples are numerically more challenging. Consider sums $S := \sum_{i=1}^{R} X_i$ of $R = 111$ i.i.d. random variables X_i following a Weibull distribution

$$W(x) := P[X_i > x] = \exp(x^\beta). \tag{21}$$

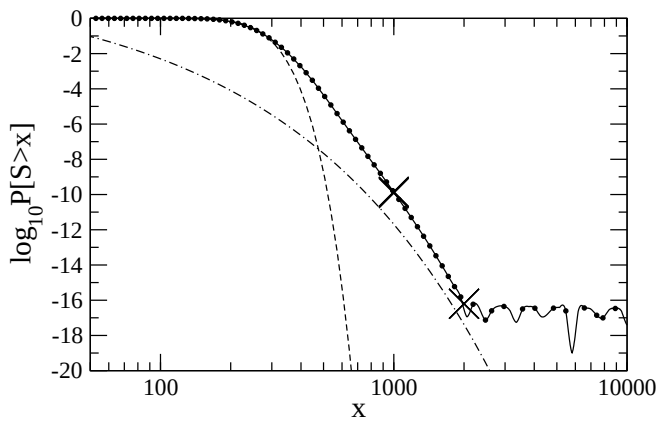

FIGURE 3: Complementary c.d.f. of the sum S of 111 independent Weibull r.v. with shape parameter $\beta = 0.5$, as computed by the method described here (for negative numerical results, the absolute value is shown). Dots correspond to the numerical grid, the solid line is a 5th order spline interpolation on logarithmic scales. The dashed line corresponds to a normal approximation, the dash-dotted line to the tail approximation valid when the sum is dominated by the single largest term. Simulation results are marked by $\times$. Five independent results were obtained at $x = 1000$ and $x = 2000$, which are here visually indistinguishable.

First, c.d.f. corresponding to a shape parameter $\beta = 0.5$ is computed. Parameters for the numerical Laplace transform are chosen as $-y_0 = y_1 = 250$, $N = 2^{13}$ and $\epsilon = 10^{-50}$. The tail of the complementary c.d.f. of the sum is shown in Fig. 3. For $S > 2000$, numerical errors (aliasing) dominate the result. The size of these errors ($< 10^{-16}$) indicates the numerical accuracy achieved. Near its mean value 222, the distribution of the sum is well described by a normal distribution (standard deviation 47.1, dashed in Fig. 3). For large S it approaches the asymptote $P[S > x] \sim RW(x)$ (dash-dotted in Fig. 3). To check if the result is correct also between these limits, the conditional Monte Carlo method proposed by Asmussen and Kroese [5] was used. Specifically, $P[S_i > x]$ was estimated by the average of 1000 pseudo-random realizations of

$$R \cdot W\left[\max\left(X_1, \ldots, X_{R-1}, x - \sum_{i=1}^{R-1} X_i\right)\right]. \tag{22}$$

As a consistency check, five realizations of this average were sampled. At $x = 1000$ values ranged from $1.17 \cdot 10^{-10}$ to $1.47 \cdot 10^{-10}$, while the Laplace method yields $1.35 \cdot 10^{-10}$. Similarly good agreement was achieved at $x = 2000$ (Fig. 3), thus confirming

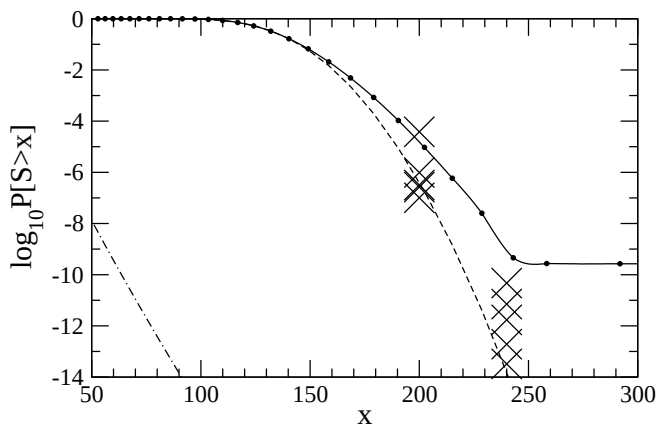

FIGURE 4: Complementary c.d.f. of the sum S of 111 Weibull r.v. with shape parameter $\beta = 0.8$. Other details are as in Fig. 3. Compared to Fig. 3, the simulation results ($\times$ at $x = 200$ and $x = 240$) become inaccurate here.

that the complementary c.d.f. is accurate down to probabilities of 10^{-16}.

Next, the c.d.f. for shape parameter $\beta = 0.8$ was evaluated. With the same values for y_0, y_1, N and ϵ as above, errors from aliasing are now $\approx 10^{-9}$. In the range $x = 150$ to $x = 240$, as shown in Fig. (4), neither the normal approximation nor the asymptotic tail formula are reliable. Additionally, Monte Carlo simulations based on Eq. (22) (again averaged over 1000 samples) now scatter strongly. Equation (22) is known to become inefficient for large x when $\beta > 0.585$ [5]. An independent verification of the c.d.f. found here for $\beta = 0.8$ might therefore be difficult.

3.4. Waiting times in M/P/1 queues

Finally, I take up one of the numerical examples used recently to demonstrate a new method for computing waiting-time distributions in M/G/1 queues with heavy-tailed service time distributions [22]. In the example considered here, the service times X_i follow a unit Pareto distribution of the form

$$\int_0^x B(t)\,dt = P[X_i < x] = 1 - (x+1)^{-\alpha}. \tag{23}$$

This case is, for example, important for questions of network design [10]. The Laplace transforms of the distributions of waiting time $W(t)$ and service time $B(t)$ are known to be related as (e.g., [15])

$$[\mathcal{L}W](s) = \frac{(1-\rho)s}{s - (1 - [\mathcal{L}B](s))\lambda}, \tag{24}$$

where λ is the arrival rate and $\rho = \lambda \mathsf{E}X$ the load. Formally, Eq. (24) corresponds to a mixture of sums of X_i with a geometric distribution of the number of terms. Equation (24) can be used to compute $W(t)$. The challenge [22] is to compute the cumulative waiting time distribution $F(t) = \int_0^t W(\tau)d\tau$ in the interval $0 \leq t \leq 250$ to an accuracy of 0.0005 for parameters α and λ as given in Table 3.4.

Two points need attention before applying the method introduced here to this problem: First, since $[\mathcal{L}B](s) \to 1$ as $s \to 0$, small inaccuracies in $\mathcal{L}B$ will produce artifacts when computing $\mathcal{L}W$ from Eq. (24). These artifacts were here removed by correcting the values of $[\mathcal{L}W](s)$, obtained numerically on the grid $s = s_k$ (with $k < j \leftrightarrow s_k < s_j$) as follows: iterating from large to small s, $[\mathcal{L}W](s_i)$ was set to $[\mathcal{L}W](s_{i+1})$ if it was smaller than $[\mathcal{L}W](s_{i+1})$ and to 1 if it was larger than 1.

Second, the transformations suggested in Sec. 3.1 to obtain periodic functions for the FFTs fail for the back-transformation if applied to directly $[\mathcal{L}W](s)$, because this function converges to $(1-\rho)$ for $-y = \log(s) \to \infty$, and not to zero. One way to fix this is to follow Ref. [22] in going over to the waiting time distribution conditional to a non-idle queue $W_b(t) = W(t)/\rho$ (for $t > 0$). This yields $[\mathcal{L}W_b](s) = \{[\mathcal{L}W](s) - (1-\rho)\}/\rho$, which has the desired limits. A corresponding relation holds for the c.d.f..

The required accuracy is comfortably reached by setting $-y_0 = y_1 = 15$, $N = 2^7 = 128$, and $\epsilon = 10^{-30}$ for all three cases of Tab. 3.4. A fifth order spline was used to interpolate between grid points. The numerical results mostly coincide with Ref. [22]. Only for $F(100) = 0.983$ in Problem 1 the last digit differs in Ref. [22]. In calculations with more ambitious parameter settings, one obtains $F(100) = 0.98316\ldots$, confirming the present result.

Marked differences can be found when comparing computation times. Calculations here where performed using a general-purpose mathematical command language on an UltraSPARC IV+ 1.5 GHz CPU. Times given in Tab. 3.4 are adjusted to 1 GHz clock rate. Contrary to Ref. [22], no attempts were made to invert the Laplace transform for individual values of t. Instead, FFTs were used to compute convolutions and deconvolutions as above, which yields $F(t)$ over the full range of $t = e^y$. Depending on

	Problem 1	Problem 2	Problem 3
λ	1	0.866666	0.8161616
α^{a}	2.25	2.083333	2.020202
Current Work:			
$F(30)$	0.904	0.854	0.830
$F(100)$	0.983	0.964	0.952
CPU time$^{\mathrm{b}}F(t)^{\mathrm{c}}$	0.6	0.6	0.6
Reference [22]:			
$F(30)$	0.904	0.854	
$F(100)$	0.984	0.964	
CPU time$^{\mathrm{b}}F(30)$	4	9	
CPU time$^{\mathrm{b}}F(100)$	2.5	5.5	
CPU time$^{\mathrm{b}}F(t)^{\mathrm{d}}$	4.5	22	35

[a] chosen such that $\rho \approx 0.8$

[b] time in seconds, normalized to 1 GHz clock frequency

[c] 128 points, geometrically spaced between $3.1 \cdot 10^{-7}$ and $3.3 \cdot 10^{6}$

[d] 25 points, linearly spaced between 0 and 250

Table 1: Parameters, numerical results and computation times for the c.d.f. $F(t)$ of waiting times in M/P/1 queues.

the problem, the improvement in computation time over the already highly efficient methods of Ref. [22] was seven to 58 fold (Tab. 3.4). However, to be fair, one has to point out that the algorithms used there automatically determined the parameters required to reach a specified accuracy, while this was done by trial and error here. Refinements of the methods of Ref. [22], for example by using continued fraction representations of the forward transform [2], might also be possible.

4. Concluding discussion

The results of the foregoing section demonstrate that the method to compute and invert Laplace transforms proposed here can efficiently be applied to problems in prob-

ability theory. The method is particularly powerful in situations involving distributions with heavy tails, which are naturally represented on a logarithmic scale.

The most important application of the method is probably the computation of the c.d.f. or sums, or mixtures of sums, of independent random variables. But correlated lognormal variables can be handled as well, provided the correlation can be factored out of the sum. This is, for example, the case for the sum $S = \sum_{i=1}^{R} \exp(\xi_i)$ where the ξ_i follow a multivariate normal distribution such that $c = \operatorname{cov}(\xi_i, \xi_j)$ is the same for all for $i \neq j$ and $\operatorname{var} \xi_i \geq c$ for all i. In an insurance setting, this would correspond to the plausible assumption that, when the claim of client i was K times larger than usual, the claims of other clients can be predicted to be typically by a factor $cK/\operatorname{var} \xi_i$ higher than usual. To evaluate the c.d.f. of S, note that S has the same distribution as $\exp(Y) \times \sum_{i=1}^{R} \exp(X_i)$ for independent normally distributed X_i and Y such that $\mathsf{E}X_i = \mathsf{E}\xi_i$, $\mathsf{E}Y = 0$ and $\operatorname{var} X_i = \operatorname{var} \xi_i - c$, $\operatorname{var} Y = c$. The c.d.f. of S can be obtained by computing first the c.d.f. of $\sum_{i=1}^{R} \exp(X_i)$ (where R may be a random or constant), and then incorporating the multiplication with $\exp(Y)$ as an addition on logarithmic scales, i.e., by a convolution of the distributions of the logarithms. Another type of sums of correlated lognormals with a factorisable structure is $\sum_{i=1}^{R} \exp\left(\sum_{j=1}^{i} \xi_i\right) = (\cdots((e^{\xi_1}+1)\, e^{\xi_2}+1)\, e^{\xi_3} \cdots)$ with independent, normal ξ_i. Such expressions occurs in certain applications in finance, where the ξ_i represent relative changes in the value of an asset over subsequent time intervals.

Since the computationally most expensive steps of the method proposed here are the two FFTs to compute the Laplace transform and the two FFTs for the backward transform, the computational complexity of the algorithm increases as $N \log N$ with the number of mesh points. All Laplace inversion algorithms working on linear scales appear to be at least of order N^2. Open problems are how to choose optimal parameters for the numerical scheme, and how this affects the complexity of the algorithm.

References

[1] Abate, J. and Valkó, P. P. (2004). Multi-precision Laplace transform inversion. *International Journal for Numerical Methods in Engineering* **60,** 979–993.

[2] Abate, J. and Whitt, W. (1999). Computing Laplace transforms for numerical inversion via continued fractions. *INFORMS Journal on Computing* **11,** 394–405.

[3] Abate, J. and Whitt, W. (2006). A unified framework for numerically inverting Laplace transforms. *INFORMS Journal on Computing* **18,** 408–421.

[4] Abu-Dayya, A. and Beaulieu, N. C. (1994). Outage probabilities in the presence of correlated lognormal interference. *IEEE Trans. Veh. Technol.* **43,** 164–173.

[5] Asmussen, S. and Kroese, D. P. (2006). Improved algorithms for rare event simulation with heavy tails. *Adv. Appl. Probab.* **38,** 545–558.

[6] Beaulieu, N. C., Abu-Dayya, A. A. and McLane, P. J. (1995). Estimating the distribution of a sum of independent lognormal random variables. *IEEE Trans. Commun.* **43,** 2869–2873.

[7] Beaulieu, N. C. and Rajwani, F. (2004). Highly accurate simple closed-form approximations to lognormal sum distributions and densities. *IEEE Commun. Lett.* **8,** 709–711.

[8] Beaulieu, N. C. and Xie, Q. (2004). An optimal lognormal approximation to lognormal sum distributions. *IEEE Trans. Veh. Technol.* **53,** 479–489.

[9] Dubner, H. and Abate, J. (1968). Numerical inversion of Laplace transforms by relating them to the finite Fourier cosine transform. *JACM* **15,** 115–123.

[10] Duffield, N. G. and Whitt, W. (2000). Network design and control using on/off and multilevel source traffic models with heavy-tailed distributions. In *Self-Similar Network Traffic and Performance Evaluation*. ed. K. Park and W. Willinger. Wiley, New York ch. 17, pp. 421–445.

[11] Dufresne, D. (2004). The log-normal approximation in financial and other computations. *Adv. in Appl. Probab.* **36,** 747–773.

[12] Frisch, U. and Sornette, D. (1997). Extreme deviations and applications. *Journal de Physique II* **7,** 1155–1171.

[13] Gaver, D. P. (1966). Observing stochastic processes and approximate transform inversion. *Operations Research* **14,** 444–459.

[14] Gradshtein, I. and Ryzhik, I. (1980). *Tables of integrals, series and products* 2nd ed. Academic Press, New York.

[15] Gross, D. and Harris, C. M. (1998). *Fundamentals of Queueing Theory* 3rd ed. John Wiley, New York.

[16] Leipnik, R. R. (1991). On lognormal random variables: I – the characteristic function. *J. Austral. Math. Soc. Ser. B* **32,** 327–347.

[17] Lopez-Fidalgo, J. and Sanchez, G. (2005). Statistical criteria to establish bioassay programs. *Health Physics* **89,** 333–338.

[18] Milevsky, M. A. and Posner, S. E. (1998). Asian options, the sum of lognormals, and the reciprocal gamma distribution. *Journal of Financial and Quantitative Analysis* **33,** 409–422.

[19] Nielsen, N. (1906). *Handbuch der Theorie der Gammafunktion.* Teubner, Leipzig.

[20] Romeo, M., Costa, V. D. and Bardou, F. (2003). Broad distribution effects in sums of lognormal random variables. *Eur. Phys. J. B* **32,** 513–525.

[21] Rossberg, A. G., Amemiya, T. and Itoh, K. (2007). Accurate and fast approximations of moment-generating functions and their inversion for log-normal and similar distributions. `http://axel.rossberg.net/paper/Rossberg2007c.pdf`, under review.

[22] Shortle, J. F., Brill, P. H., Fischer, M. J., Gross, D. and Masi, D. M. B. (2004). An algorithm to compute the waiting time distribution for the M/G/1 queue. *INFORMS Journal on Computing* **16,** 152161.

[23] Slimane, S. B. (2001). Bounds on the distribution of a sum of independent lognormal random variables. *IEEE Trans. Commun.* **49,** 975–978.

[24] Stehfest, H. (1970). Algorithm 368: numerical inversion of Laplace transforms. *Commun. ACM* **13,** 47–49.

[25] Talbot, A. (1979). The accurate inversion of Laplace transforms. *J. Inst. Maths. Applics.* **23,** 97–120.

[26] Valkó, P. and Abate, J. (2004). Comparison of sequence accelerators for the Gaver method of numerical Laplace transform inversion. *Computers and Mathematics with Applications* **48,** 629–636.

[27] Valkó, P. P. and Vojta, B. L. The list. `http://www.pe.tamu.edu/valko/public_html/NIL/LapLit.pdf` 2003. accessed Jun 1, 2007.

[28] Vallade, M. and Houchmandzadeh, B. (2003). Analytical solution of a neutral model of biodiversity. *Phys. Rev. E* **68,** 061902.